Mathematics

HP COMPUTER CURRICULUM

Number Set

STUDENT LAB BOOK

HEWLETT PACKARD

HEWLETT hp PACKARD

Hewlett-Packard
Computer Curriculum Series

mathematics
STUDENT LAB BOOK

number sets

by E. A. Herber
Golden High School
Golden, Colorado

edited by Jean Danver and Christine Doerr
Hewlett-Packard

Hewlett-Packard Company
11000 Wolfe Road
Cupertino, California 95014

HP 5951-4441

March 1973

The Hewlett-Packard Computer Curriculum Series represents the published results of a Curriculum Development project sponsored by the Data Systems Division of Hewlett-Packard Company. This project is under the directorship of Jean H. Danver.

This material is designed to be used with any Hewlett-Packard system with the BASIC programming language such as the 9830A Educational BASIC, and the 2000 and 3000 series systems.

TABLE OF CONTENTS

1 INTRODUCTION

3 LIST OF SYMBOLS
3 Flow Chart Symbols
4 Algebraic Notation and Equivalent BASIC Language Symbols
4 Number Set Designations

5 NUMBERS AND MATHEMATICS

7 INTEGERS
7 Greatest Common Divisor
7 Exercise 1 – Finding the Greatest Common Divisor
9 Euclidean Algorithm
10 Exercise 2 – Using the Euclidean Algorithm
11 Prime Numbers
11 Exercise 3 – Prime Numbers
13 Exercise 4 – Determining Prime Numbers by the "Sieve of Eratosthenes" Method
15 Perfect Numbers
15 Exercise 5 – Determining Perfect Numbers
16 Pythagorean Triples
16 Exercise 6 – Finding Pythagorean Triples
18 Suggested References for this Section

19 RATIONAL NUMBERS
19 Exercise 7 – Denumeration of the Set of Positive Rational Numbers
21 Exercise 8 – Forming Rational Numbers, Using All Integers in a Given Real Number Interval
23 Exercise 9 – Rational Numbers Formed From a Random Set of Integers
25 Suggested References for this Section

27 IRRATIONAL NUMBERS
27 Newton's Method for Finding Square Root
27 Exercise 10 – A Rational Approximation of $\sqrt{n}$, n Rational >0
29 Incrementing Method
29 Exercise 11 – Incrementing Method for Finding $\sqrt{n}$, n Rational >0
31 Pi, the Most Famous Irrational Number
32 Exercise 12 – Computing Π by a Probability Method
33 Suggested References For This Section

INTRODUCTION

This Mathematics Student Book was written to enrich your study of mathematics by showing you how to use a computer as a modeling device. The computer is particularly helpful in quickly performing the repetitious steps of algorithms, thus making mathematical investigations easier and more exciting. You will write computer programs which will help you to understand the major concepts involved in the study of a particular mathematical area. If you become more involved in investigating the laws of mathematics, this book will have achieved its aim.

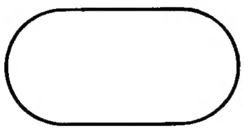
To use the Student Book for Number Sets, you will need the following: First, you should have one year's background in algebra. Second, the book assumes that you already know how to write a program in the BASIC programming language, and that you understand programming techniques for inputting data, performing algebraic operations, designating variables, assigning values to variables, looping, and printing results. If you do not have this background, you will want to study BASIC before attempting this material. Consult the BASIC Manual for the computer you use. Last, in order to complete the exercises in this Student Book, you will need to have adequate access to a computer (about two hours per week for a terminal system). If more time is available, you may be able to experiment further on your own, either to improve your programs or to investigate other areas of mathematics that interest you.

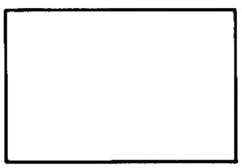
Each section of this book is organized in the same way. First, the mathematical concepts needed to complete the exercises are reviewed. References are listed at the end of each section in case you want to study these concepts in greater detail. Next, each exercise is presented. Finally, an approach is suggested in the Problem Analysis and a flow chart is included to illustrate this approach. The suggested approach was chosen because it brings out the concepts which are being stressed, but the program can sometimes be written more efficiently. Once you have completed the exercise by following the logic in the flow chart, you are encouraged to rewrite the program using more sophisticated programming techniques. You might also want to impose more conditions on the problem to make it more interesting to solve.

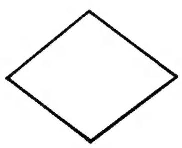
There are many different ways to solve one problem by programming. Experiment and learn as you go. You'll find you are learning something new each time, both about your subject matter and about using the computer to solve problems and model mathematical algorithms.

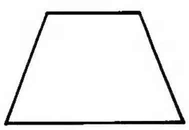
LIST OF SYMBOLS

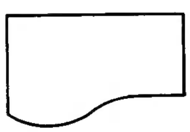
FLOW CHART SYMBOLS

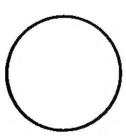
 ------------ Start or Stop

 ------------ Defines a process

 ------------ Represents a decision point

 ------------ Represents computer input

 ------------ Represents computer output

 ------------ Used to connect one part of a flow chart continued at some other place

ALGEBRAIC NOTATION AND EQUIVALENT *BASIC* LANGUAGE SYMBOLS

Algebraic Notation	*BASIC Notation*	*Meaning*
$+$	+	Addition
$-$	–	Subtraction
$\cdot$ or $\times$	*	Multiplication
$\div$ or /	/	Division
$\sqrt{X}$	SQR(X)	Square root of X
$\lvert Y \rvert$	ABS(Y)	Absolute value of Y
$[\![X]\!]$	INT(X)	Greatest integer less than or equal to X
$=$	=	Equals
$\neq$	# or <>	Does not equal
$<$	<	Less than
$>$	>	Greater than
$\leqslant$	<=	Less than or equal to
$\geqslant$	>=	Greater than or equal to
$\rightarrow$	=	Replaced by
( ) or [ ]	( ) or [ ]	Inclusive brackets or parentheses
A_i	A(I)	Subscripted variable
$A_{i,j}$	A(I,J)	Double subscripted variable
(None)	RND(X)	Assign a random number to the variable X

NUMBER SET DESIGNATIONS

N – Natural number set
W – Whole number set
I – Set of integers

Q – Rational number set
Z – Irrational number set
R – Real number set

NUMBERS AND MATHEMATICS

It may seem obvious to you that numbers are a necessary element of mathematical study. As a matter of fact, they have contributed to the development of mathematics in two important ways:

First, numbers are used to express quantitative relationships between objects or concepts to which number values have been assigned. In this role, they have provided the very basis for organized mathematical study. Numbers have served this utilitarian role in business, economics, science, engineering, etc. In recent years, the computer has made it possible to apply numerical methods to other fields such as medicine, science, and social systems.

Second, number theory (the study of numbers) is an important area of mathematics. From the beginning of the use of numbers, man has been curious about them. This curiosity first emerged as a form of mysticism ascribed to numbers. But as time went on, the interest in the properties of numbers developed into a serious study of their underlying principles. The study of number theory was not undertaken for practical reasons, but it has been found to have many practical applications.

INTEGERS

"God created the natural numbers; everything else is man's handiwork."
–– Leopold Kroeneckes (1823–1891)

The properties of the set of integers have fascinated mathematicians for hundreds of years. The Pythagoreans, a group of mathematicians led by Pythagoras (572-501 B.C.), probably did the first serious study of the integers. Their investigation included the study of certain series of integers, factors, prime numbers, etc. Euclid (300 B.C.) extended the study to include number pairs.

The computer has made it possible for mathematicians to greatly extend the scope of all previous studies of integers. For example, the Pythagoreans investigated perfect numbers. Euclid went on to develop a formula, $2^{n-1}(2^n - 1)$, that yields a perfect number when $2^n - 1$ is a prime number. Yet by 1876, only twelve perfect numbers had been found, mainly because of computational limitations. The twelfth perfect number is $2^{126}(2^{127} - 1) = 137{,}438{,}691{,}328$. You can see why further calculations were abandoned until the computer became available. There are now 23 known perfect numbers. The 23rd is $2^{11{,}212}(2^{11{,}213} - 1)$, which has 6,751 digits!

Integers are basic not only to the real number system but to the actual design of the digital computer. Therefore, it is appropriate to begin our study of number sets with them.

GREATEST COMMON DIVISOR

There are many arithmetic operations on the set of integers for which we need to find the greatest common divisor. Let's define the greatest common divisor (g.c.d.) in the following way: if d is a common divisor of a set of integers, and every other common divisor of that set is a divisor of d, then d is the *greatest common divisor* (g.c.d.). One method for finding the g.c.d. of a set is educated guesswork. Start with the smallest integer in the set. If it is a common divisor for all other members of the set, it is the g.c.d. Otherwise, factor it and test each of its factors until you find one which is a common divisor of the set. If there are no factors left greater than the common divisor found, it is your g.c.d. If there are larger untested factors, test them one by one until you are left with the g.c.d. If there is no common divisor for the set, the integers in the set are said to be *relatively prime.*

EXERCISE 1 – Finding the Greatest Common Divisor

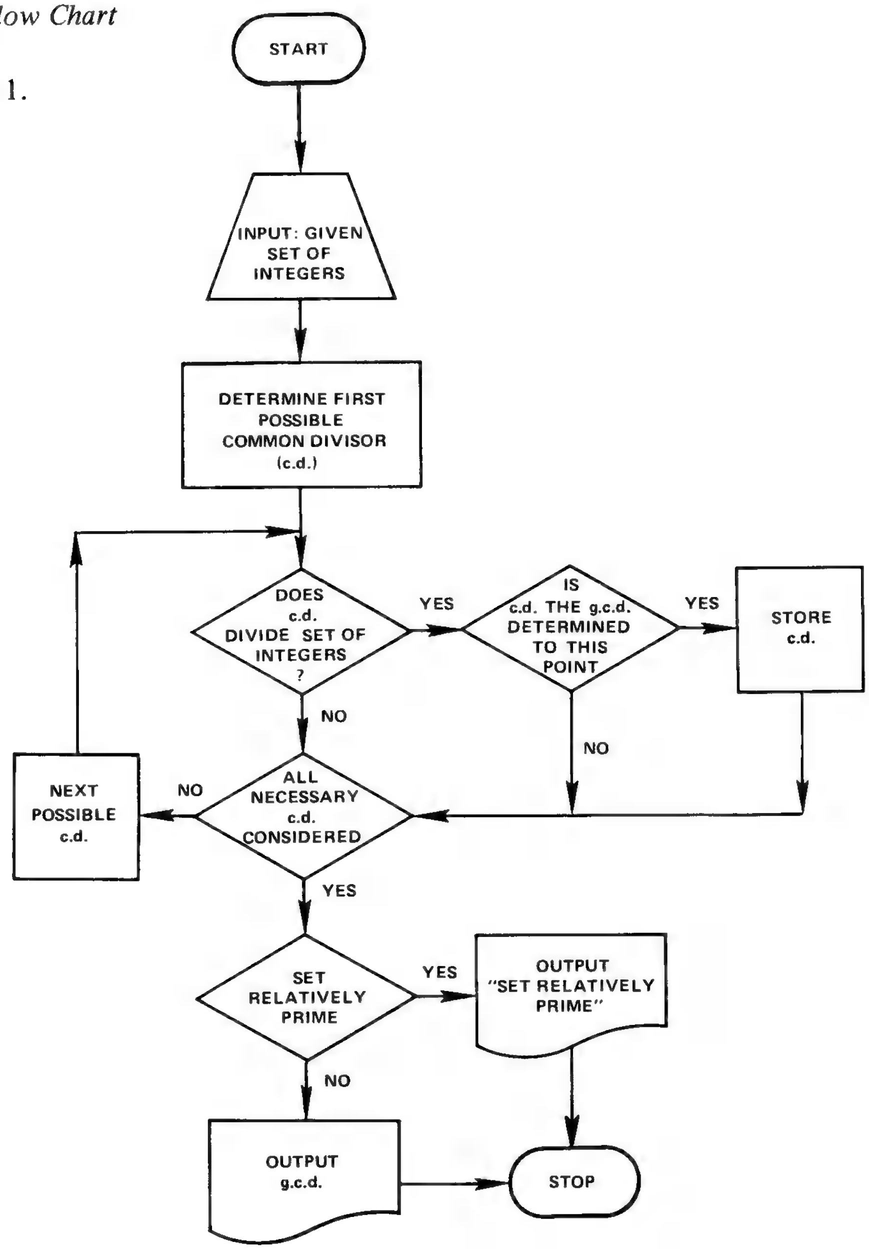
Write a computer program that will find the greatest common divisor (g.c.d.) of a given set of positive integers. Test your program on the sets G = {91,26,169,286} and F = {71,251,149,353}. If the elements of the set are relatively prime, have the computer so indicate.

Problem Analysis

The flow chart given here uses *uneducated* guesswork as a method for solving this problem. It tests all integers from 1 to some element of the set as common divisors. Once you have written the program described by it, make a flow chart of the more efficient method described above and alter your program accordingly.

Macro Flow Chart

Exercise 1.



EUCLIDEAN ALGORITHM

Euclid devised an algorithm for finding the greatest common divisor of two positive integers. This *Euclidean Algorithm*, as it is called, is a sequence of steps applying the division algorithm.

The division algorithm tells us that for any integers a and b, there exists integers q and r such that a = b(q) + r or in another form, a − b(q) = r. For this discussion, we will accept this division algorithm without proof. Given two integers, the Euclidean Algorithm substitutes the integers for a and b in the division algorithm. The variable q is given the value of the greatest integer for which b(q) < a and then the value of r is found. a is then replaced with b and b with r. This procedure is repeated until r equals zero.

The final value of b is the g.c.d. of the original values of a and b. To continue to check the elements of the set, we use the g.c.d. just found for a and the next element in the set for b. We continue the procedure until all elements in the set have been considered. Now let's apply the algorithm to the set below:

$$A = \{81,54,1458,36\}$$

To find the g.c.d. of 81 and 54, the first two elements of A, we follow our procedure:

$$a \leftarrow 81,\ b \leftarrow 54 \text{ gives us } 81 - 54(1) = 27.$$

The next step gives us a ← 54, b ← 27 and 54 − 27(2) = 0. Therefore, the g.c.d. is this final value of b, or 27.

The next element in the set is 1458. We start our calculation with a equal to the g.c.d. already found, 27 in this case, and b equal to 1458:

$$27 - 1458(0) = 27$$
$$a \leftarrow 1458,\ b \leftarrow 27$$
$$1458 - 27(54) = 0$$

Therefore: The g.c.d. of 27 and 1458 is 27.

a ← 27, b ← 36 (last integer in set)
27 − 36(0) = 27
a ← 36, b ← 27
36 − 27(1) = 9
a ← 27, b ← 9
27 − 9(3) = 0

Therefore: The g.c.d. of the integers in Set A is 9.

EXERCISE 2 — Using the Euclidean Algorithm

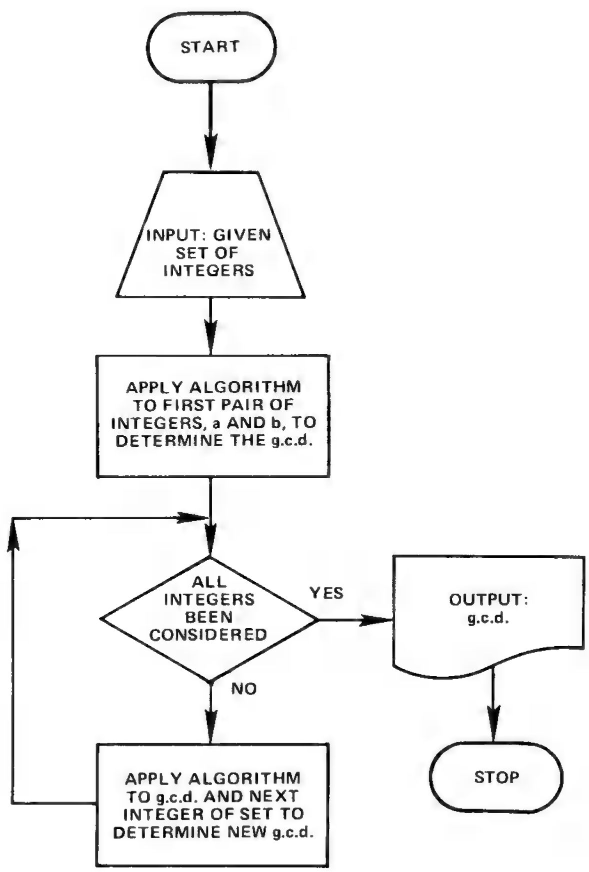
Write a computer program which uses the Euclidean Algorithm to find the greatest common divisor of a set of integers. Test your program on sets G and F of Exercise 1.

Problem Analysis

Once you understand the above example, you should have no trouble completing this exercise.

Macro Flow Chart

Exercise 2.



PRIME NUMBERS

A *prime number* is a positive integer which has no divisor other than itself and the number one. The most obvious method for determining primeness is simply trial-and-error. We'll use this method for the next exercise.

EXERCISE 3 – Prime Numbers

(a) Write a computer program that will determine all prime numbers between 1 and 500.

(b) After you have a working program for Part (a), modify your program so it will count the number of prime numbers in the intervals 1–100, 1–200, . . . 1–500.

(c) Adjust the program for Part (a) to count and print twin primes from 1 to n as n ← 100, 200, . . . 500. Twin primes are consecutive primes. For example, 3 and 5 are twin primes.

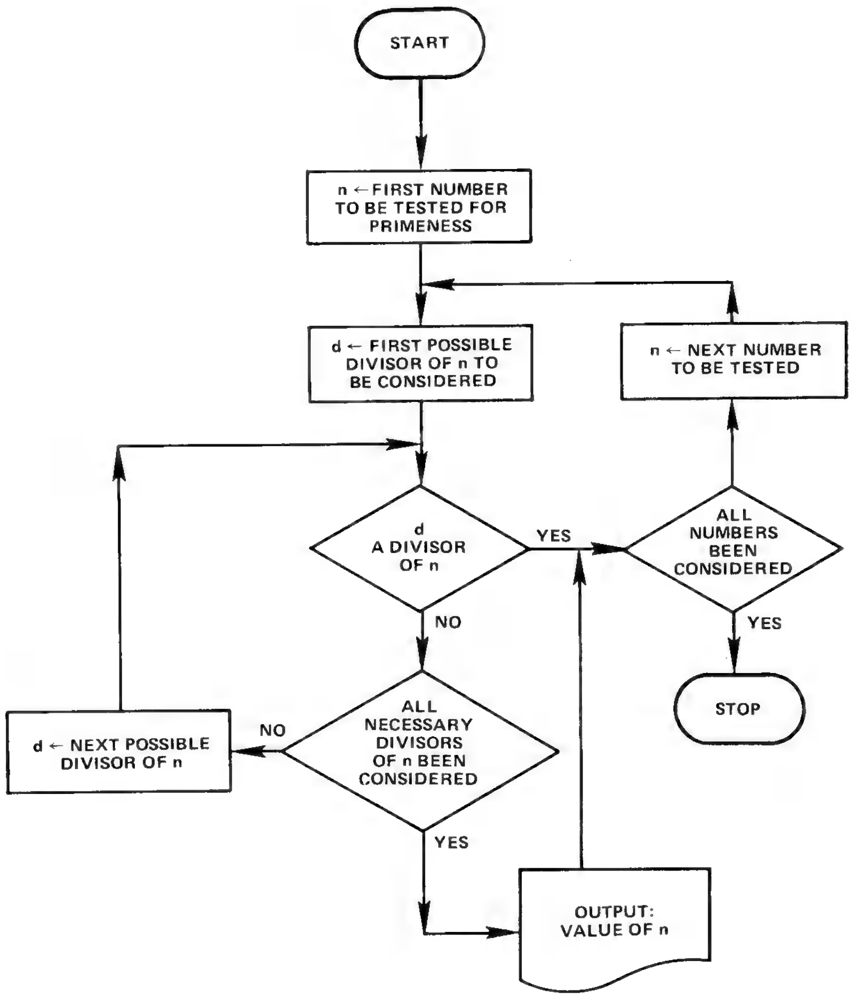
(d) Adjust program 3(a) to have the computer print out ratios n/m and $n/\log_e n$ for $n \leftarrow 10^2, 10^3, 10^4, 10^5, 10^6$, where m is the number of primes less than n.

Problem Analysis

The flow chart clearly illustrates the solution for Part (a). No flow charts are included for Parts (b), (c), and (d) since they are merely modifications of Part (a).

Macro Flow Chart

Exercise 3(a)



EXERCISE 4 – Determining Prime Numbers by "Sieve of Eratosthenes" Method

Write a computer program to determine all prime numbers from 1 to some number n using the "Sieve of Eratosthenes" approach.

Problem Analysis

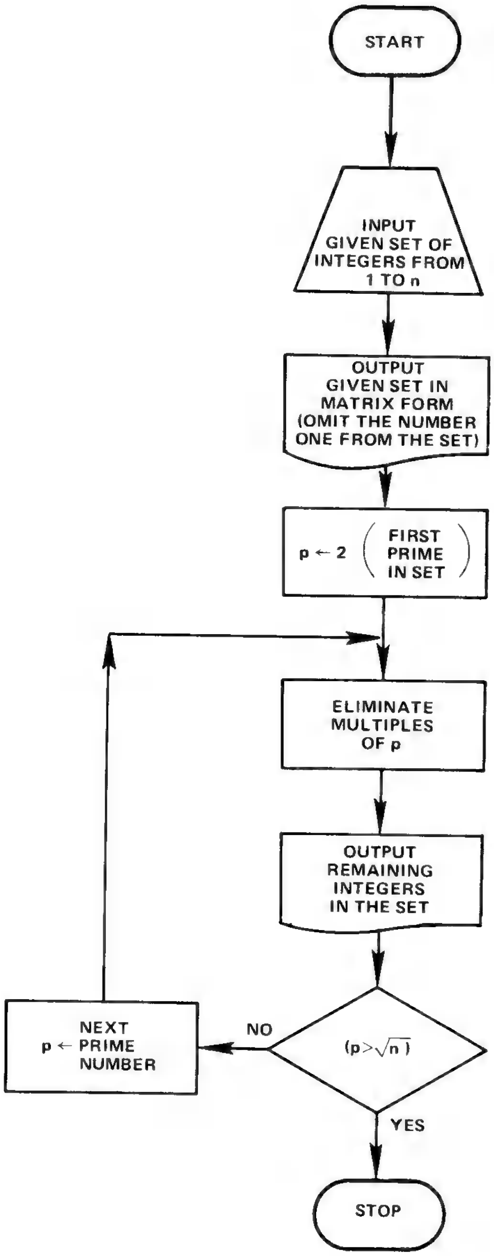
The sieve approach involves picking the first prime number in the set and then eliminating all multiples of that prime. This is repeated until only prime numbers remain. For example, we'll use this approach to find all prime numbers between 1 and 36. We've shown the set under consideration and the output generated after each step.

Input:	2	3	4	5	6	7	8
	9	10	11	12	13	14	15
	16	17	18	19	20	21	22
	23	24	25	26	27	28	29
	30	31	32	33	34	35	36
First Output:	2	3		5		7	
(Eliminating	9		11		13		15
multiples of		17		19		21	
2 except 2)	23		25		27		29
		31		33		35	
Second Output:	2	3		5		7	
(Eliminating			11		13		
multiples of		17		19			
3 except 3)	23		25				29
		31				35	
Third Output:	2	3		5		7	
(Eliminating			11		13		
multiples of		17		19			
5 except 5)	23						29
		31					

The steps illustrated lead us to construct the following flow chart.

Macro Flow Chart

Exercise 4.



PERFECT NUMBERS

A number, n, is called a *perfect number* when the sum of all its divisors (including 1 but excluding itself) is equal to n.

EXERCISE 5 – Determining Perfect Numbers

Write a program to find all perfect numbers less than or equal to 500. Use the following output format for your results:

$$f_1, f_2 \ldots f_c, n$$

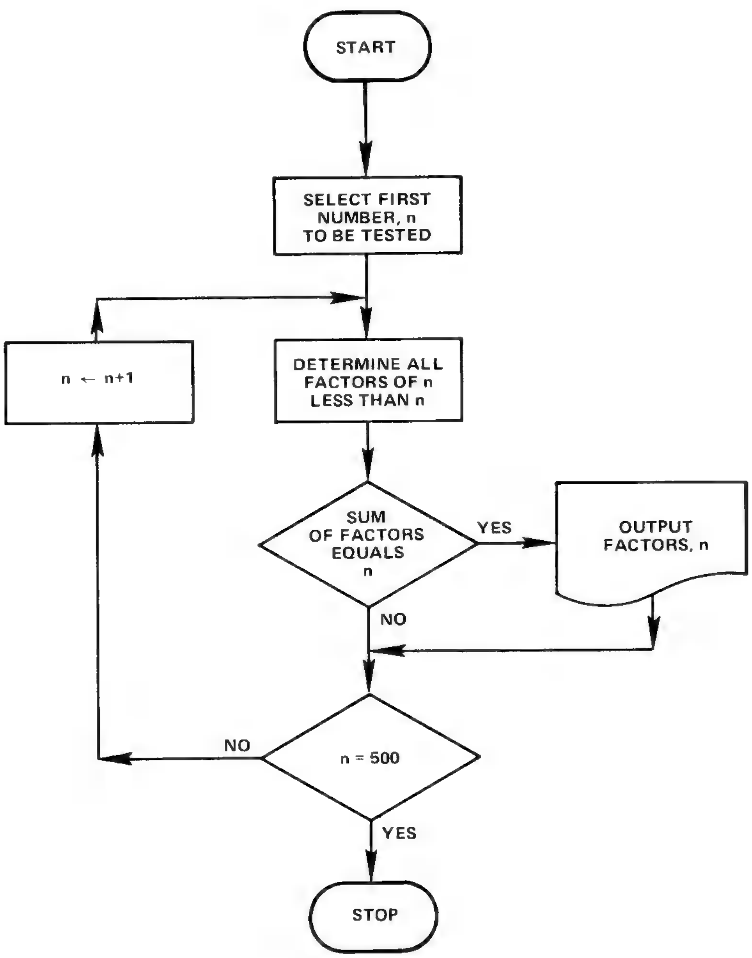
where n is the perfect number and $f_1, f_2, \ldots f_c$ *are the factors of n such that* $f_c < n$.

Problem Analysis

This problem is solved by merely testing each number from 1 to 500 to see which numbers meet the condition defined above.

Macro Flow Chart

Exercise 5.



CAN YOU FIND THE NEXT ONE?

According to *Tecnica Education Corporation Newsletter*, December 1970, eighteen-year-old Roy Ferguson of Dallas, Texas recently developed a computer program that determined the 21st perfect number. The number has 5834 digits.

PYTHAGOREAN TRIPLES

Positive integer solutions to the Pythagorean equation $z^2 = x^2 + y^2$ can be found by using the relationships (1) $x = m^2 - n^2$, (2) $y = 2mn$, and (3) $z = m^2 + n^2$ where m and n are positive integers with $m > n$. These solutions are called *Pythagorean triples.*

EXERCISE 6 – Finding Pythagorean Triples

(a) *Using the relationships defined above write a computer program to find 100 Pythagorean triples satisfying $z^2 = x^2 + y^2$, with x, y, and z positive integers.*

(b) *Adjust your program so that it prints only* primitive *triples, i.e., triples that have no common divisors greater than 1.*

Problem Analysis

(a) Our task is to find 100 ordered pairs, (m,n,), with $m > n$, so that the values of x, y, and z obtained by using the given relationships satisfy the condition $z^2 = x^2 + y^2$.

Since $m > n > 0$ we begin with

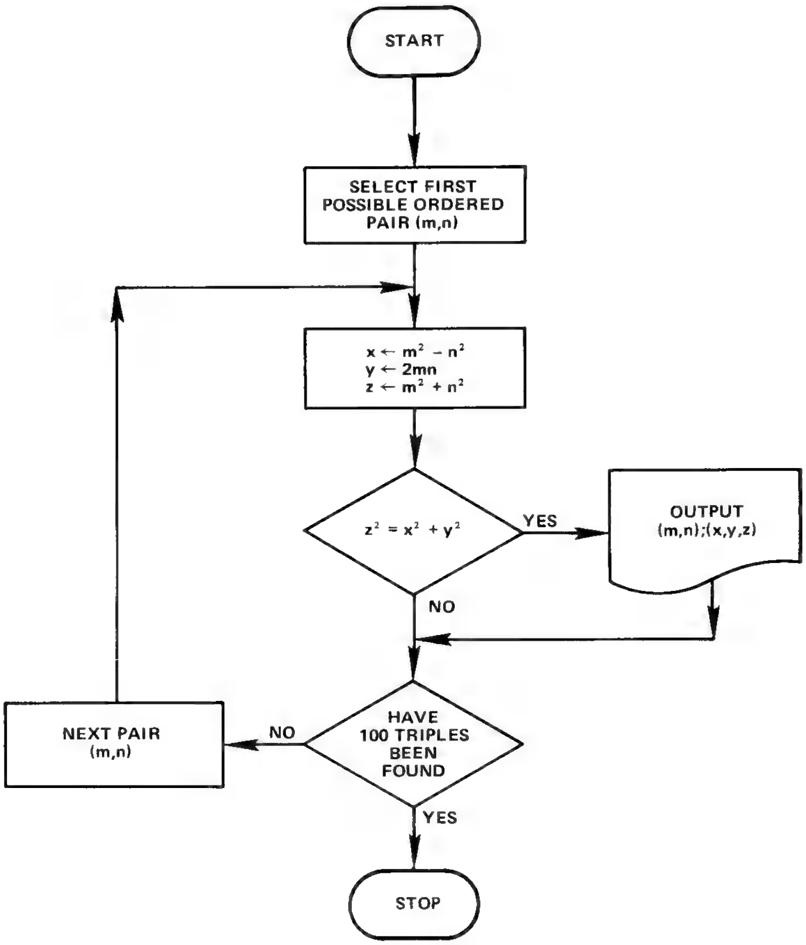
First	(m,n) = (2,1)
Next	(m,n) = (3,1)
Next	(m,n) = (3,2)
Next	(m,n) = (4,1)
.	.
.	.
.	.

The flow chart is clear from this point on.

(b) Your program will only need to determine that x and y are relatively prime. Can you verify that this will assure that x, y, and z have no common divisors except 1?

Macro Flow Chart

Exercise 6(a)



SUGGESTED REFERENCES FOR THIS SECTION

Allendoerfer, Carl B., and Oakley, Cletus O., *Principles of Mathematics*, 2nd Edition, McGraw-Hill, New York, 1963.

Dolciani, Mary P., et al., *Modern Introductory Analysis*, Houghton-Mifflin Co., Boston, 1964.

Edwards, Ronald R., *Introduction to the Theory of Numbers*, Houghton-Mifflin Mathematics Enrichment Series, Boston, 1968.

Gamow, George, *One, Two, Three . . . Infinity*, The Viking Press, New York, 1954.

Jacobs, Harold R., *Mathematics, A Human Endeavor*, W.H. Freeman & Co., San Francisco, 1970.

Keedy, M.L., et al., *Algebra and Trigonometry*, Holt, Rinehart and Winston, New York, 1967.

McCoy, Neil H., *Introduction to Modern Algebra*, Allyn and Bacon, Inc., Boston, 1960.

Computer Oriented Mathematics, An Introduction for Teachers, NCTM, Washington D.C., 1965.

RATIONAL NUMBERS

A *rational number* is a number which can be expressed as a/b, where a and b are integers with b ≠ 0. The development of rational numbers was prompted by the need to measure quantities which were smaller than whole units.

Early man was slow to develop a number system that included numbers representing "aliquot parts" (fractions) of a unit. For a long time, it was considered adequate to measure quantities to the nearest unit. The Egyptians treated only fractions with a numerator of 1, and the Babylonians did not master fractional numbers until 2000 B.C.

Abstractly, the need for rational numbers is justified by the need to be able to solve equations of the form ax + b = c, where a, b, and c are integers and a ≠ 0.

We will use the computer to investigate the property of denumerability of rational numbers. The set of rational numbers is said to be *denumerable* because it is "countable," i.e., it can be shown to be in one-to-one correspondence with the natural numbers. Figure 1 illustrates one way the set of positive rational numbers can be denumerated. This method is thoroughly described in Courant's, *What Is Mathematics.*

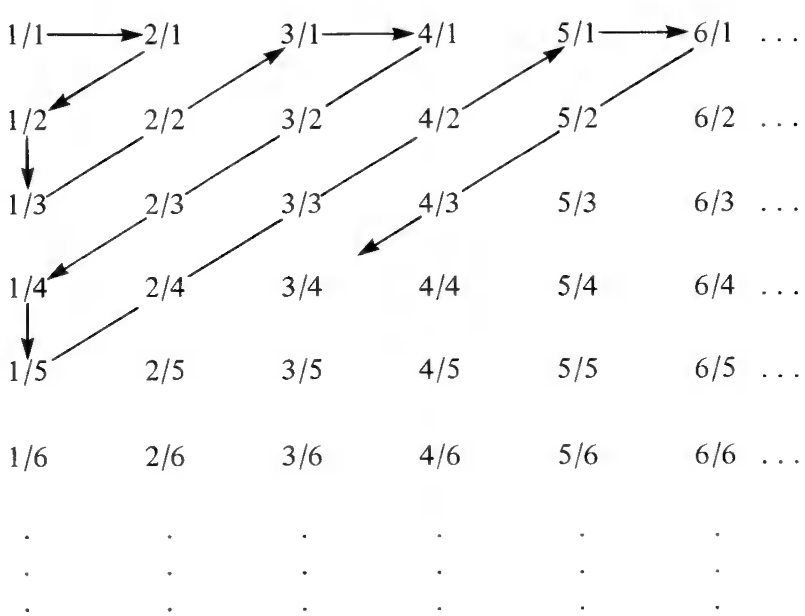


Figure 1. Denumerating the Set of Positive Rational Numbers

If you follow the path of the arrow in Figure 1, it is obvious that the positive rational numbers can be denumerated. If all numbers on the path are denumerated, some numbers will be listed more than once (2/1 and 4/2, for example).

EXERCISE 7 – Denumeration of the Set of Positive Rational Numbers

(a) Write a computer program that will print the first 100 rational numbers of the list generated as described above.

(b) Alter your program so the output includes zero and the negative rational numbers.

Problem Analysis

Our problem, then, is to develop an algorithm that will generate the above sequence of numbers, printing and counting the rational numbers 1/1, 1/2, 2/1, 1/3, . . . until 100 distinct rational numbers have been printed.

Let's examine the numbers along each diagonal of the path:

(a) 1/1
(b) 2/1, 1/2
(c) 1/3, 2/2, 3/1
(d) 4/1, 3/2, 2/3, 1/4
(e) 1/5, 2/4, 3/3, 4/2, 5/1
(f) 6/1, 5/2, 4/3, 3/4, 2/5, 1/6

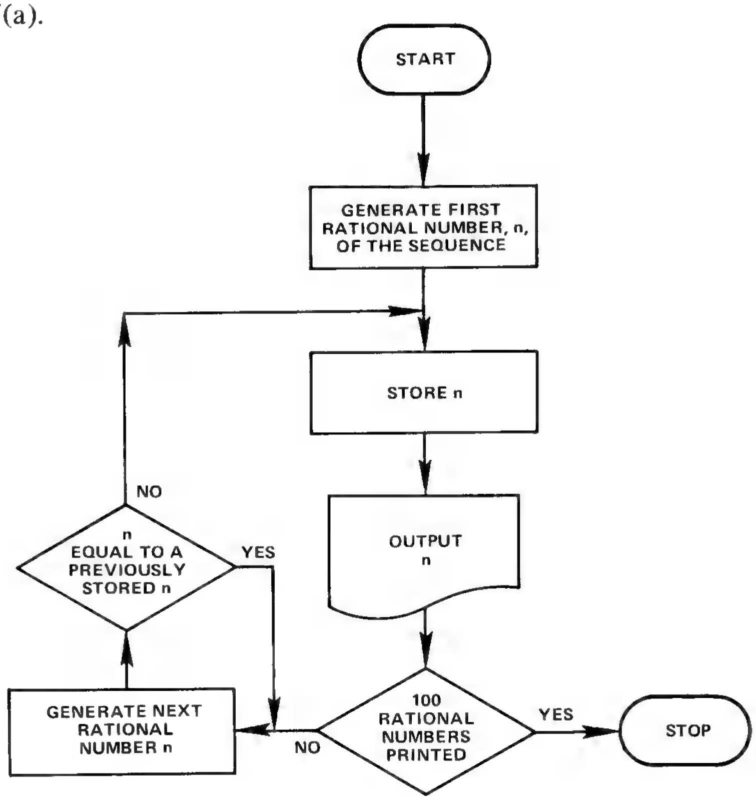
and so on.

Note that the numbers increase or decrease within each segment (diagonal) of the sequence. Changing the order of the numbers within a segment would not void the process but would simplify the programming of the process.

You should realize that the output of this algorithm will not list the rational numbers in order of magnitude, but it will illustrate the ability to produce the whole set of rational numbers, if the computer runs indefinitely.

Macro Flow Chart

Exercise 7(a).



EXERCISE 8 – Forming Rational Numbers, Using All Integers in a Given Real Number Interval

(a) *Given a set of integers,* $W = \{x \mid -7 \leqslant x \leqslant 7\}$, *have the computer print and count all meaningful rational expressions that can be formed from set W.*

(b) *Adjust your program so the computer will print only those expressions that do not duplicate values of expressions already printed. Also, have the computer output the decimal approximations of each number printed.*

(c) *Adjust the program from part (b) to print all expressions in reduced form.*

Problem Analysis

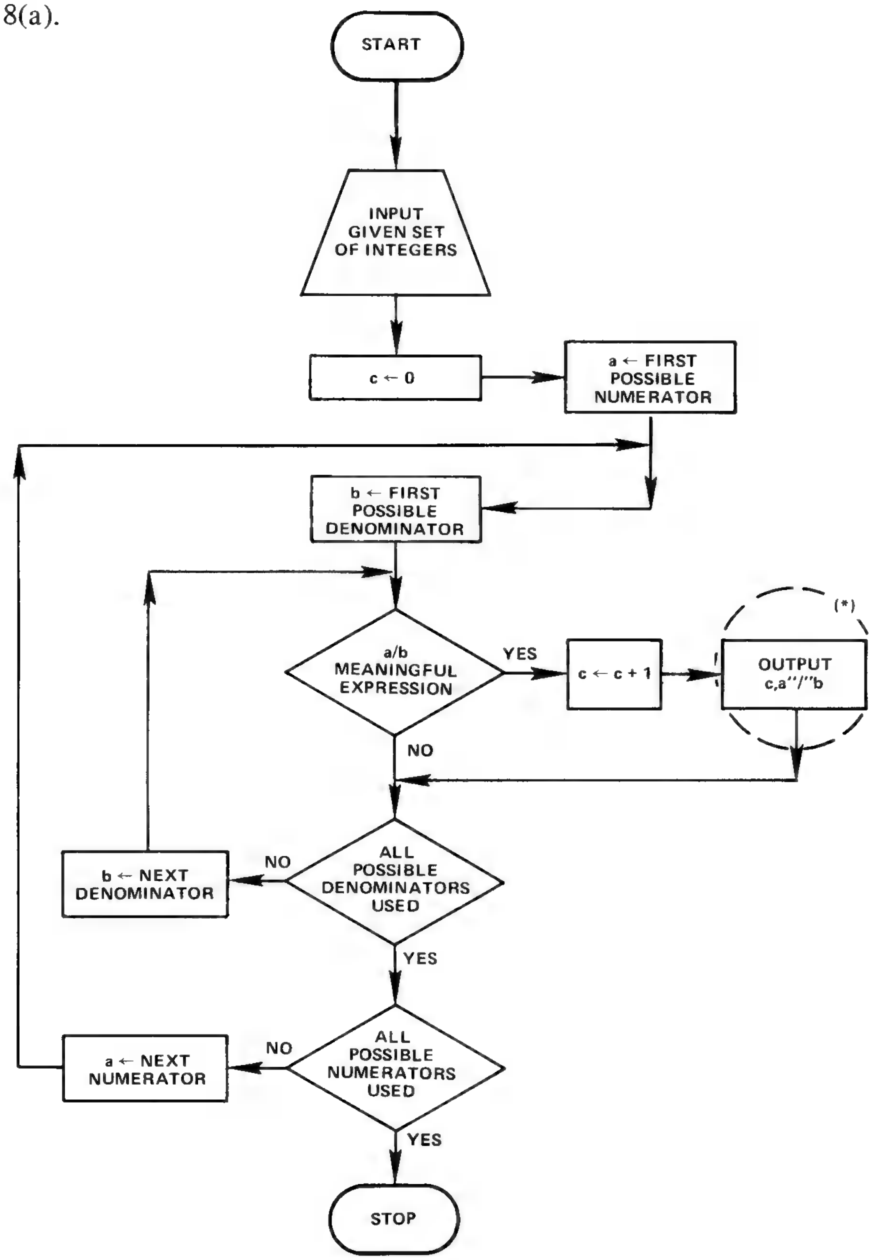
(a) If we were doing this assignment by hand, one approach might be to write:

$$-7/-7, -7/-6, -7/-5 \ldots -7/-1, -6/-7, -6/-6, -6/-5, \ldots$$
$$0/-7, 0/-6, 0/-5, \ldots 0/7 \ldots 7/7.$$

The flow chart below explains the principle behind this approach.

Macro Flow Chart

Exercise 8(a).

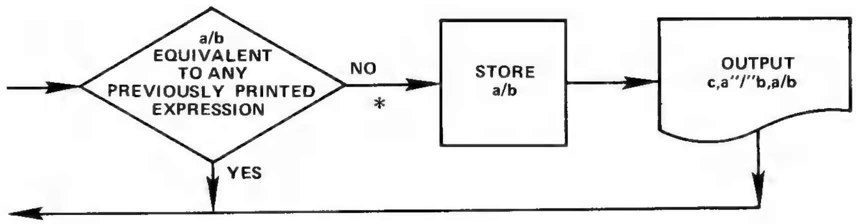


NOTES:

1. The section in the circle marked (*) will be replaced by another section to meet the conditions of Part (b).
2. a/b indicates quotient of a divided by b.
3. a"/"b represents the rational expression of a "over" b.

Part (b).

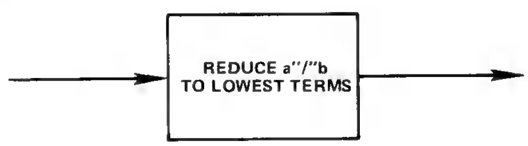
Substitute the following section for the circle in the flow chart for Part (a).



A flow chart section will be inserted at (*) to satisfy the conditions of Part (c).

Part (c)

Insert the following section as indicated into flow chart of Part (b).



EXERCISE 9 – Rational Numbers Formed from a Random Set of Integers

Given a set of random integers, $A = \{-16, 3, 45, 0, -24, -6, 7, -19, 38\}$, *have the computer form all possible unique rational numbers, separating them into three categories:*

(a) $-1 < p/q < 0$
(b) $0 < p/q < 1$
(c) $| p/q | > 1$, p and q ϵ A.

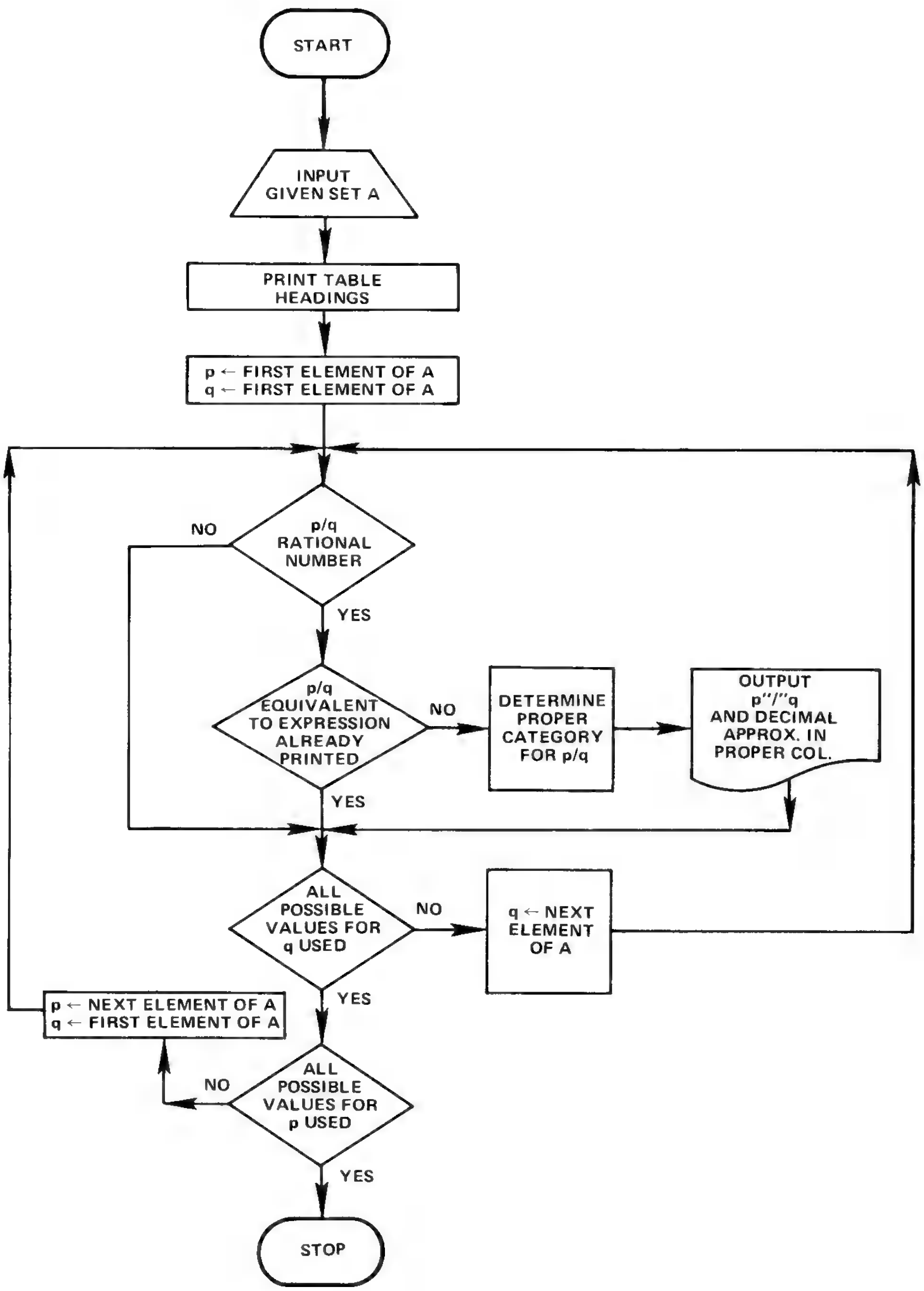
Have the computer output results in column form, printing the decimal approximation of each rational number.

Problem Analysis

This is a variation of the general case considered in Exercise 8. The procedure is the same.

Macro Flow Chart

Exercise 9.



SUGGESTED REFERENCES FOR THIS SECTION

Allendoerfer, Carl B., and Oakley, Cletus O., *Principles of Mathematics,* 2nd Edition, McGraw-Hill, New York, 1963.

Courant, Richard, and Robbins, Herbert, *What Is Mathematics,* Oxford University Press, New York, 1958.

Dolciani, Mary P., et al., *Modern Introductory Analysis,* Houghton-Mifflin Co., Boston, 1964.

Edwards, Donald R., *Introduction to the Theory of Numbers,* Houghton-Mifflin Mathematics Enrichment Series, Boston, 1968.

Singh, Jagjit, *Great Ideas of Modern Mathematics: Their Nature and Use,* Dover Publications, Inc., New York, 1959.

A Second Course in Algebra and Trigonometry with Computer Programming, Colorado Schools Computing Science Curriculum Development Project, 1969.

IRRATIONAL NUMBERS

Early mathematicians were convinced that all points on the number line are used up when a point is assigned to each rational number. About 500 B.C., the Pythagoreans proved that there are points on the line which do not correspond to rational numbers. In particular, they proved that no rational number corresponds to the measure of the diagonal of a unit square. A new category of numbers was invented, called *irrational numbers,* to represent known values which could not be expressed as rational numbers. Irrational numbers find widespread application today.

NEWTON'S METHOD FOR FINDING SQUARE ROOT

One method of determining a decimal approximation to the square root of a number is known as *Newton's method.* It is also called the successive approximation or averaging method. Whatever we call it, we start by making a guess at the square root of a number, n. We'll call this guess g_1. If $g_1 > \sqrt{n}$, you can see that $n/g_1 < \sqrt{n}$. Or, if $g_1 < \sqrt{n}$, then $n/g_1 > \sqrt{n}$. We'll assume that $g_1 > \sqrt{n}$. A logical second guess, g_2, would be the average of g_1 and n/g_1. We continue the process by picking g_3 equal to the average of g_2 and n/g_2 and so on until our guess, g_1, is within a given tolerance of the actual value of the square root of n. Figure 2 shows the first two steps on the number line.

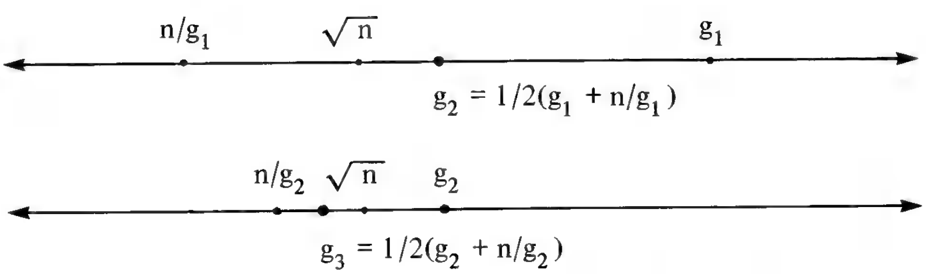


Figure 2. A Number Line Representation of Newton's Method

EXERCISE 10 – Rational Approximation of $\sqrt{n}$, n rational > 0

(a) Using Newton's method, write a computer program that will find a decimal approximation to the square root of any positive rational number accurate to within .001 of the actual value. Test your program on several numbers. Compare the results with a square root table.

(b) Adjust the program of Part (a) to find the cube root of a positive rational number. Will your program find cube roots of negative numbers?

(c) Now, you might wonder, since we are taking a cube root, wouldn't a good guess be $1/3(g_{i-1} + n/g_{i-1}{}^2)$? Try it and see what happens.

(d) We will not be able to show the development here, but it can be shown that $g_i \leftarrow 1/3(2g_{i-1} + n/g_{i-1}^2)$ is another good substitution to use to find the cube root of n. Use it in your program. Compare your results with those obtained in the first cube root program. Do some research on why the algorithm will produce cube roots.

Problem Analysis

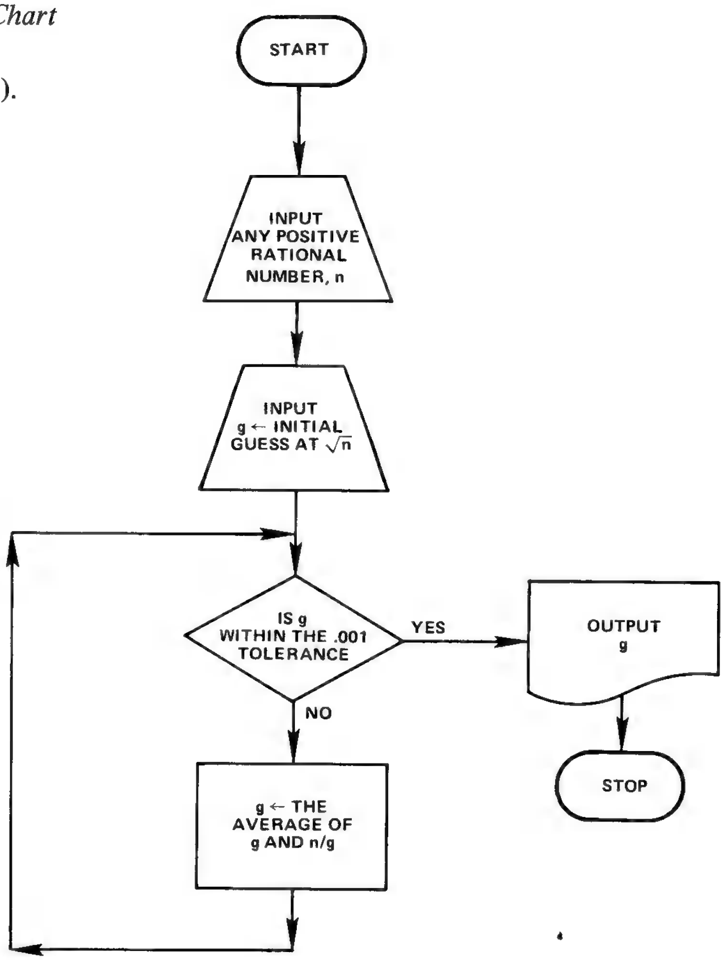
(a) The flow chart clearly follows the procedure discussed above.

(b) This part of the exercise requires some modification to our procedure. We start with a guess, g_1, at the actual value of $\sqrt[3]{n}$. Let's assume $g_1 > \sqrt[3]{n}$, which means that $n/g_1^2 < \sqrt[3]{n}$. Of course, if our guess is accurate, $g_1^3 = n$ and $n/g_1^2 = g_1$. But guesses aren't very often that accurate.

We zero in on the actual value by making g_2 equal to the average of g_1 and n/g_1^2, and so on. In general, then, $g_i = 1/2(g_{i-1} + n/g_{i-1}^2)$. Substitute this formula into your program for Part (a).

Macro Flow Chart

Exercise 10(a).



INCREMENTING METHOD

A less sophisticated method for finding the nth root of a number is the *incrementing method.* This method demonstrates the power of the computer to perform a systematic search for a desired value within a certain tolerance.

Suppose we wish to find $\sqrt[3]{76}$ within .001 of the actual value. We make a guess g, an integral value. Assume we guess g = 2, then we test g^3 and find $2^3 < 76$. Then we increment g so g = 3. g^3 is still less than 76 so we let g = 4, then 5 until we find $5^3 > 76$. Therefore, we know that $\sqrt[3]{76}$ is between 4 and 5. We set g back to 4 and increment g by .1 until $g^3 > 76$. Again, we reset g to the previous value so that $g^3 < 76$, and repeat the process with the next increment. We continue this procedure until we reach the desired degree of precision. We find that $4.235 < \sqrt[3]{76} < 4.236$, and either 4.235 or 4.236 is an acceptable solution.

We might find that the initial guess, g, was greater than $\sqrt[3]{76}$. In that case, the same procedure is used after the value of g is *decremented* until $g < \sqrt{76}$.

EXERCISE 11 – Incrementing Method for Finding $\sqrt{n}$, n rational

Write a computer program that will find a number g < rth root of a positive rational number n, so that $n - g^r < t$ where t is a given tolerance.

The output of the problem will be much more interesting if the program is written so that each tested value of g is printed. In that way, you can examine the searching process used by the computer prior to the final output.

Apply your program to the following cases:

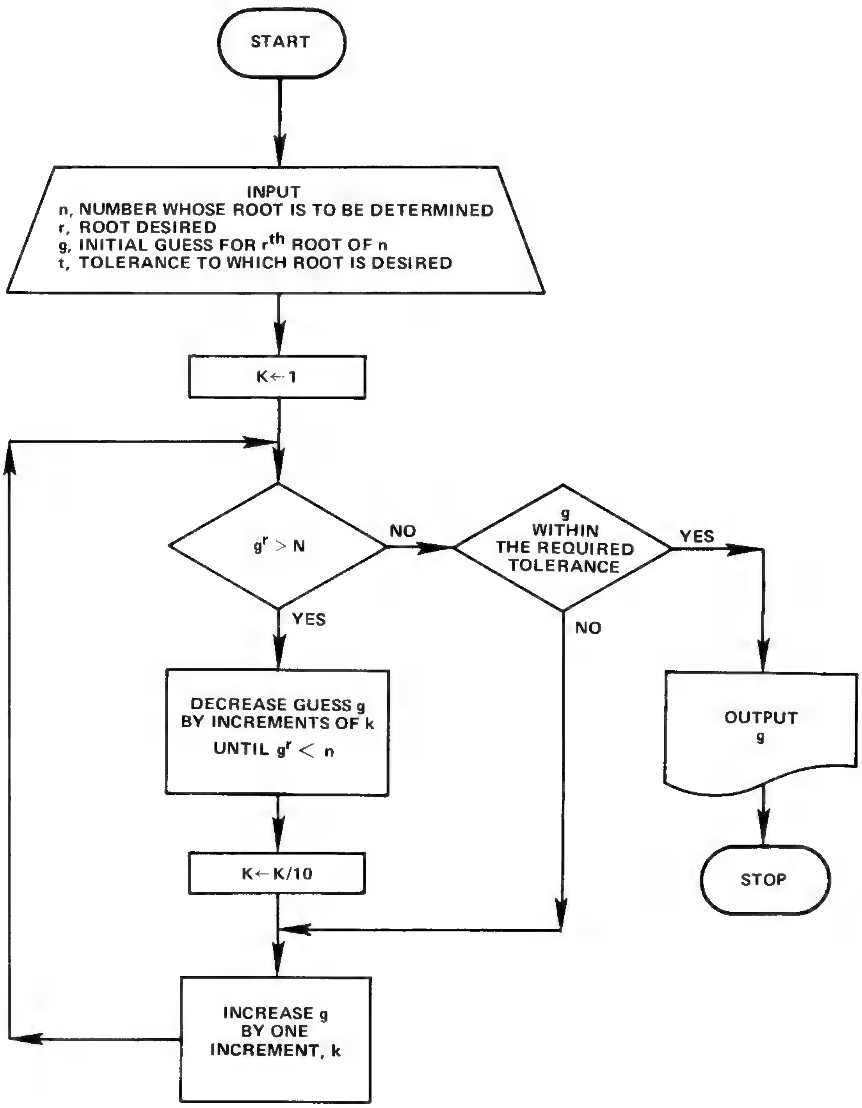
1. *cube root of 16, t = .001*
2. *fourth root of 81, t = .001*
3. *square root of .0234, t = .00001*

Problem Analysis

This problem is a very straightforward application of the incrementing method. When incrementing make the new increment equal to one tenth of the previous increment.

Macro Flow Chart

Exercise 11.



Π, THE MOST FAMOUS IRRATIONAL NUMBER

The number Π is undoubtedly the most famous irrational number. It has fascinated man since he first discovered that it existed. Since then, mathematicians have made it the subject of innumerable studies, most of them trying to determine one of the following: 1) the exact value of Π, and 2) the distribution of its digits. In fact, for centuries an unbelievable amount of effort was expended in computing Π to as many decimal places as possible.

The value of Π was first established in ancient Oriental civilizations as the ratio of the diameter to the circumference of a circle. The earliest value given to it was 3, according to Biblical reports (I Kings 7:23; II Chron. 4:2). About 240 B.C., Archimedes computed it to be between 223/71 and 22/7 or approximately equal to 3.14. By 150 A.D., it had been calculated to be 3.1416.

During the next 1500 years, many mathematicians struggled with computing Π to more and more decimal places. All used some approach related to the circle. The last major effort using this approach was made in 1630 by Grienberger, who was able to compute Π to 39 decimal places.

By the end of the 18th century, mathematicians had demonstrated that Π is not the exclusive property of circles. Various series were developed which can be used to compute Π (reference Eves' and Davis' books for details). Using such formulas, the value of Π was computed to more and more places until, in 1873, William Shanks of England computed it to 707 places. It took him 15 years to complete his calculations.

In 1940, Edward Kasner and James Newman, in their book *Mathematics and the Imagination,* stated: "even today it would require 10 years of calculation to determine Π to 1,000 places." Yet in 1949, by use of the electronic calculator ENIAC, the Army Ballistic Laboratories in Aberdeen, Maryland produced Π to 2035 places in 70 hours. Today, Π has been computed to more than 100,000 decimal places.

Two interesting approaches to computing Π are found in the area of probability. The first is the Buffon needle problem which we will not attempt to explain here. Refer to Gamow's book for an explanation of this problem. The other approach is based on a relationship of randomly chosen pairs of positive integers. We will investigate this second approach.

The relationship is as follows: the probability that two randomly chosen positive integers are relatively prime is $6/\Pi^2$. This means that if we consider an infinite number of pairs of randomly chosen numbers and tabulate those pairs that are relatively prime as successes, s, and those that are not as failures, f, then $s/(s + f) = 6/\Pi^2$. Solving this equation for Π, we get $\Pi = \sqrt{(6s + 6f)/s}$.

EXERCISE 12 – Computing Π by a Probability Method

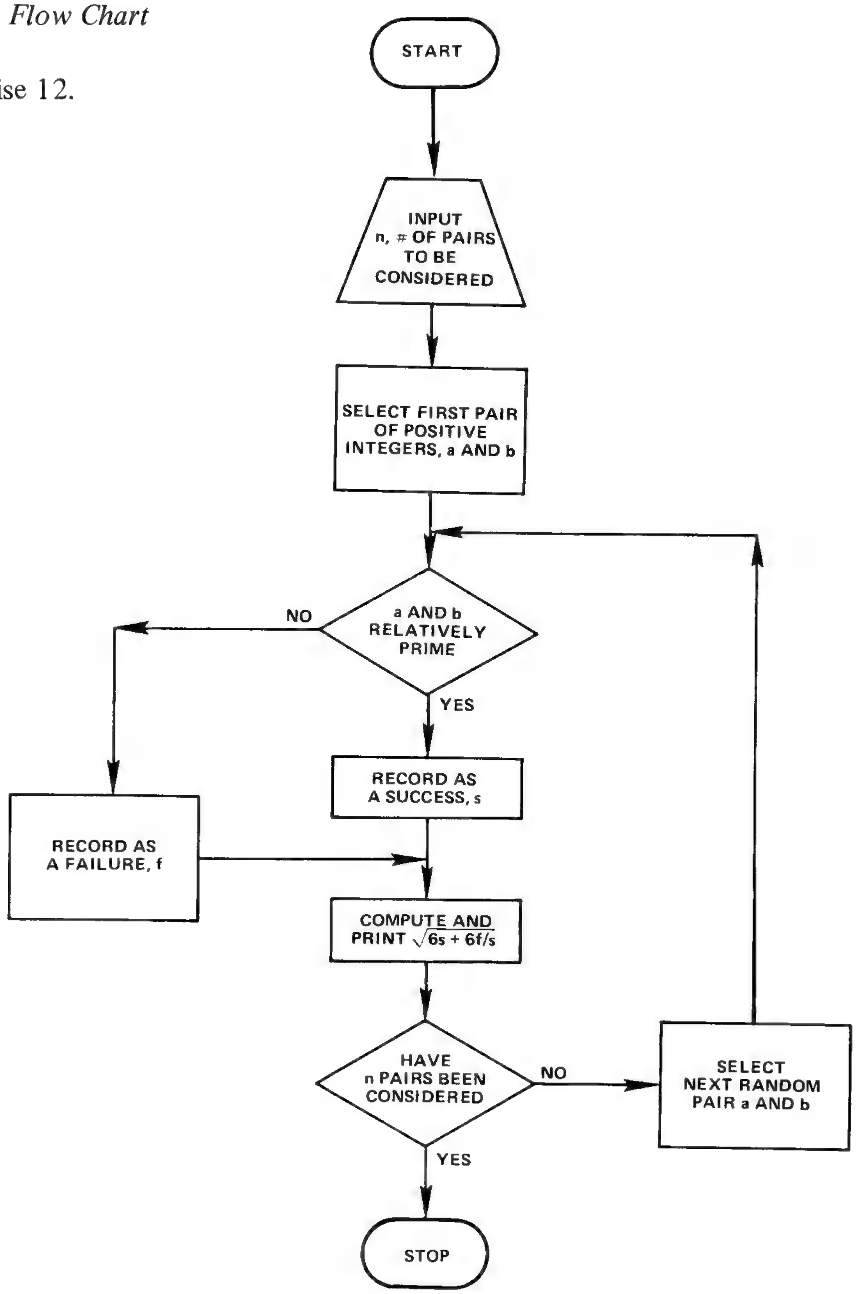
Write a computer program that will consider n pairs of random positive integers and, after each pair has been tested for relative primeness, evaluate and print $\sqrt{(6s + 6f)/s}$. Run your program for n = 100, 200 and 500.

Problem Analysis

You will be able to compute π to a surprising degree of precision by using only random numbers selected from the set of integers 1 to 1000. Of course, you can use numbers larger than 1000 if you desire.

Macro Flow Chart

Exercise 12.



SUGGESTED REFERENCES FOR THIS SECTION

Courant, Richard, and Robbins, Herbert, *What Is Mathematics,* Oxford University Press, New York, 1958.

Davis, Philip J., *The Lore of Large Numbers,* Random House, Inc., New York, 1961.

Dolciani, Mary P., et al., *Modern School Mathematics, Algebra I,* Houghton-Mifflin, Co., Boston, 1967.

Eves, Howard, *An Introduction to the History of Mathematics,* Rhinehart and Co., New York, 1953.

Gamow, George, *One Two Three . . . Infinity,* The Viking Press, New York, 1954.

Jacobs, Harold R., *Mathematics, A Human Endeavor,* W.H. Freeman and Co., San Francisco, 1970.

Rosenbloom, Paul C., and Schuster, S., *Prelude to Analysis,* Prentice-Hall, Inc., New York, 1966.

Algorithms, Computation and Mathematic, Revised Edition, School Mathematics Study Group, 1966.

NOTES

NOTES

NOTES